

Fig. 1

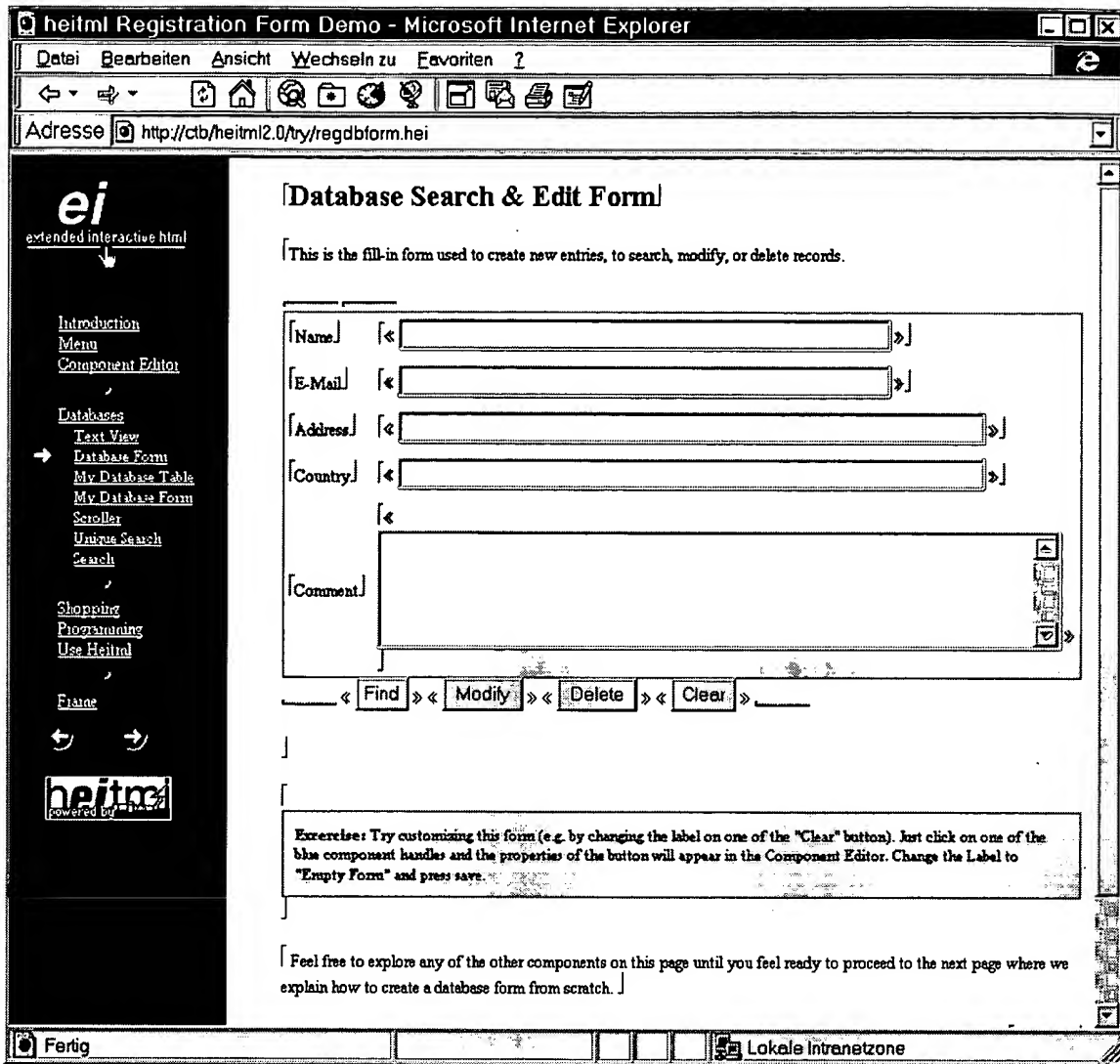


Fig. 2

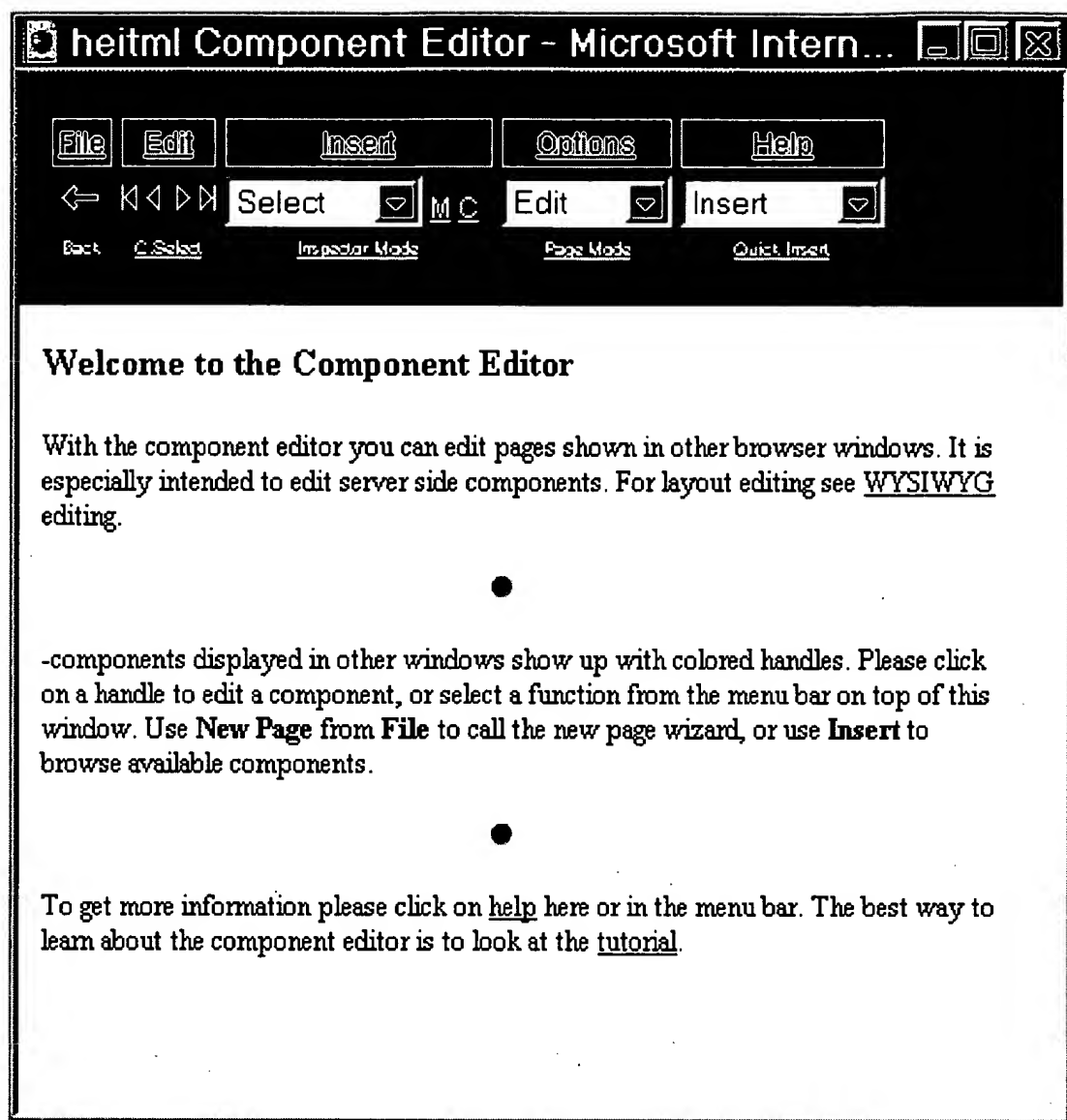


Fig. 3: Component Editor Window after Startup

hehtml Component Editor - Microsoft Internet ...

File Edit Insert Options Help

← → ⏪ ⏩ Select M C Edit Insert

Fast C Select Inspector Mode Page Mode Quick Insert

## Fieldtext

Text field.

Property	Value	Description
Name	<input type="text" value="Guest_Name"/>	Field name.
Size	<input type="text" value="50"/>	Field size in characters.
Maxlength	<input type="text"/>	Limit to the length of the field's value.
Value	<input type="text"/>	Initial field value.
Mandatory	<input type="checkbox"/>	Check to require the user to fill out the field.
Trim	<input checked="" type="checkbox"/>	Check to trim leading and trailing white-space from value.
Descr	<input type="text"/>	A description of the field value.
Disabled	<input type="radio"/> Yes <input type="radio"/> No <input checked="" type="radio"/> Like form	Is field disabled?
Password	<input type="checkbox"/>	Check to render input unreadable.

Save Delete Cancel Source

Fig 4: Component Editor Window displaying a Component Edit Page



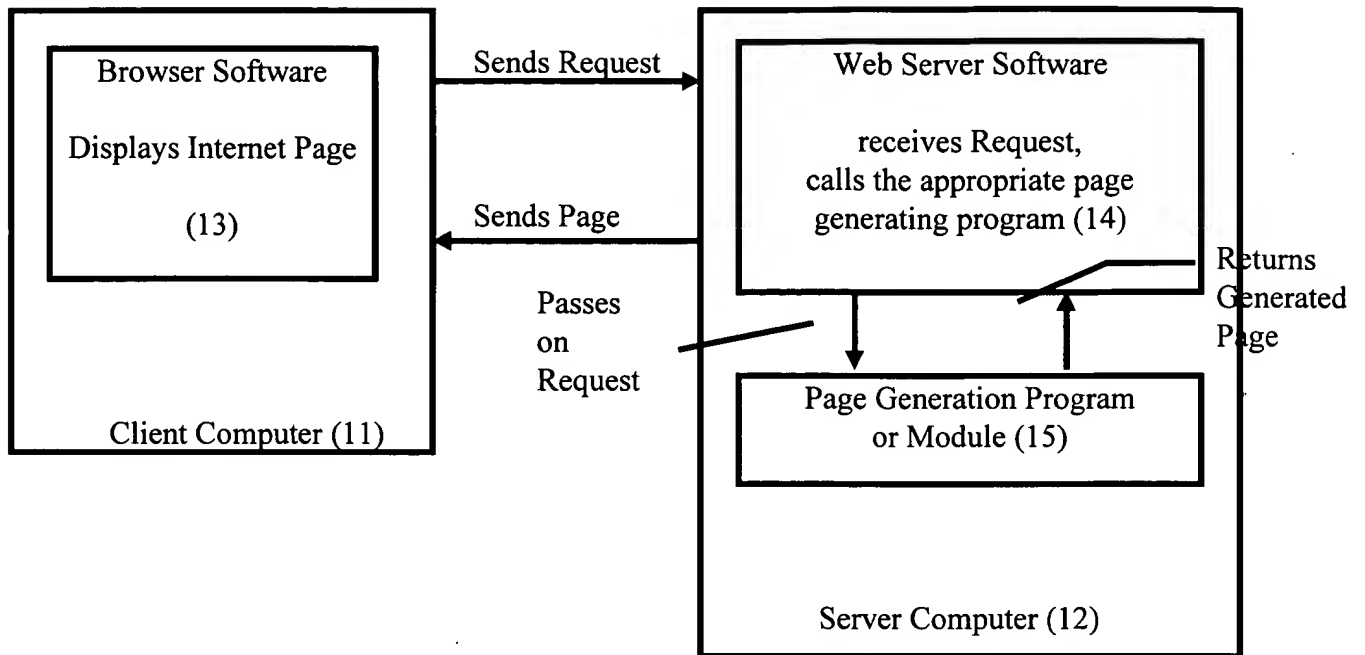


Fig. 6: State of the Art Model for Server Based Internet Applications

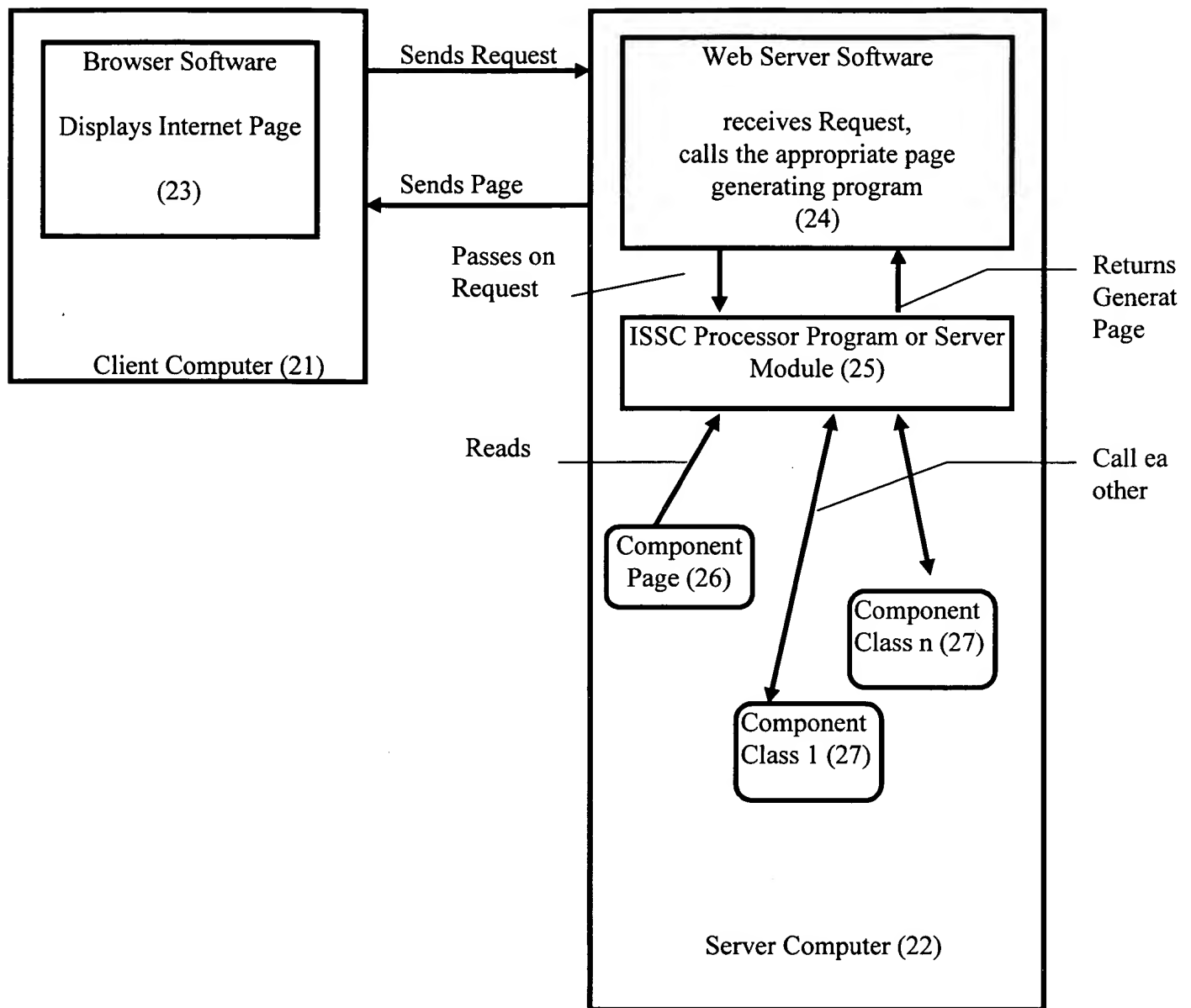


Fig. 7: Model for Server Based Internet Applications with ISSCs

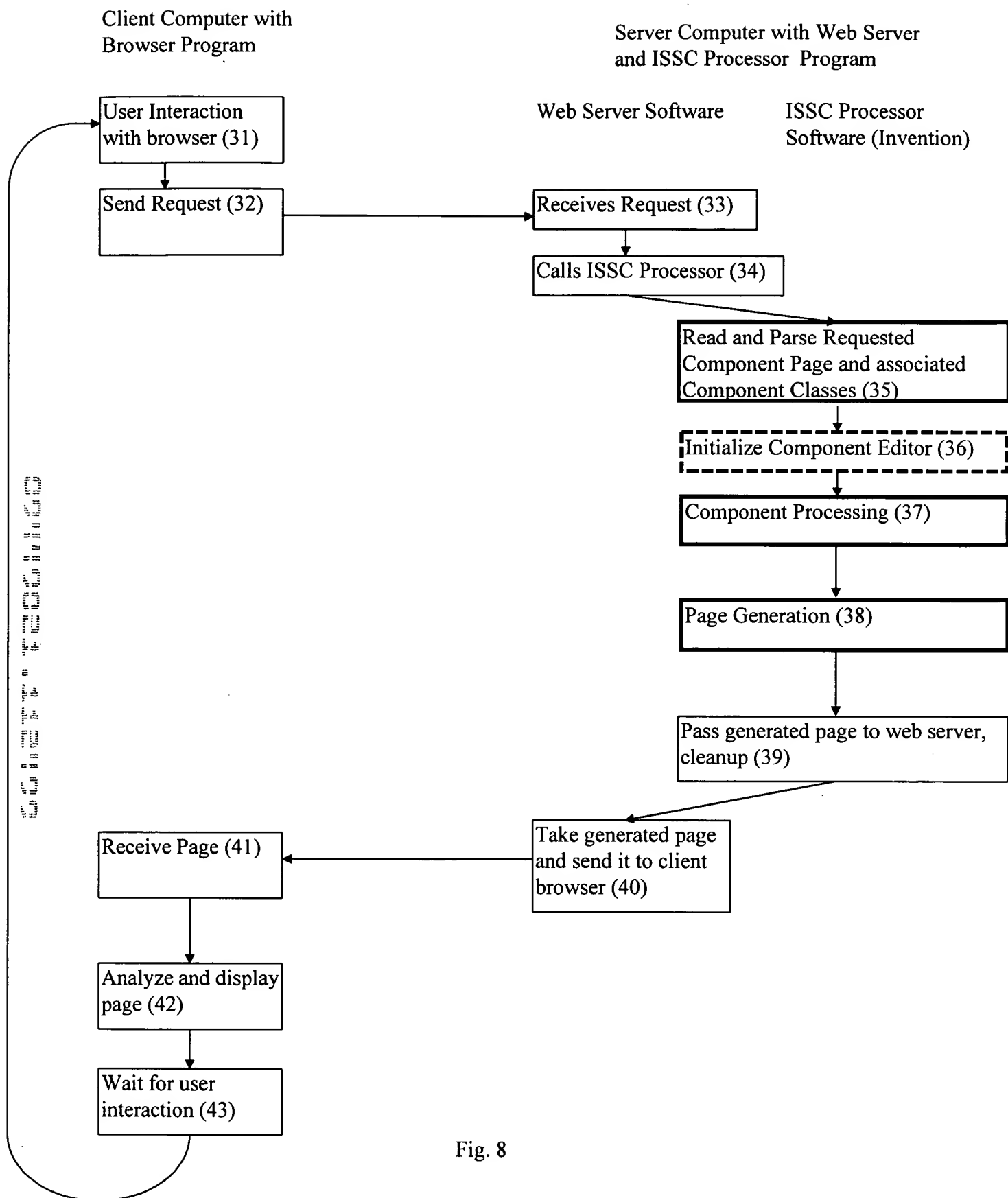


Fig. 8



## Generation Algorithm

Parameter  $l$  is a cb-list

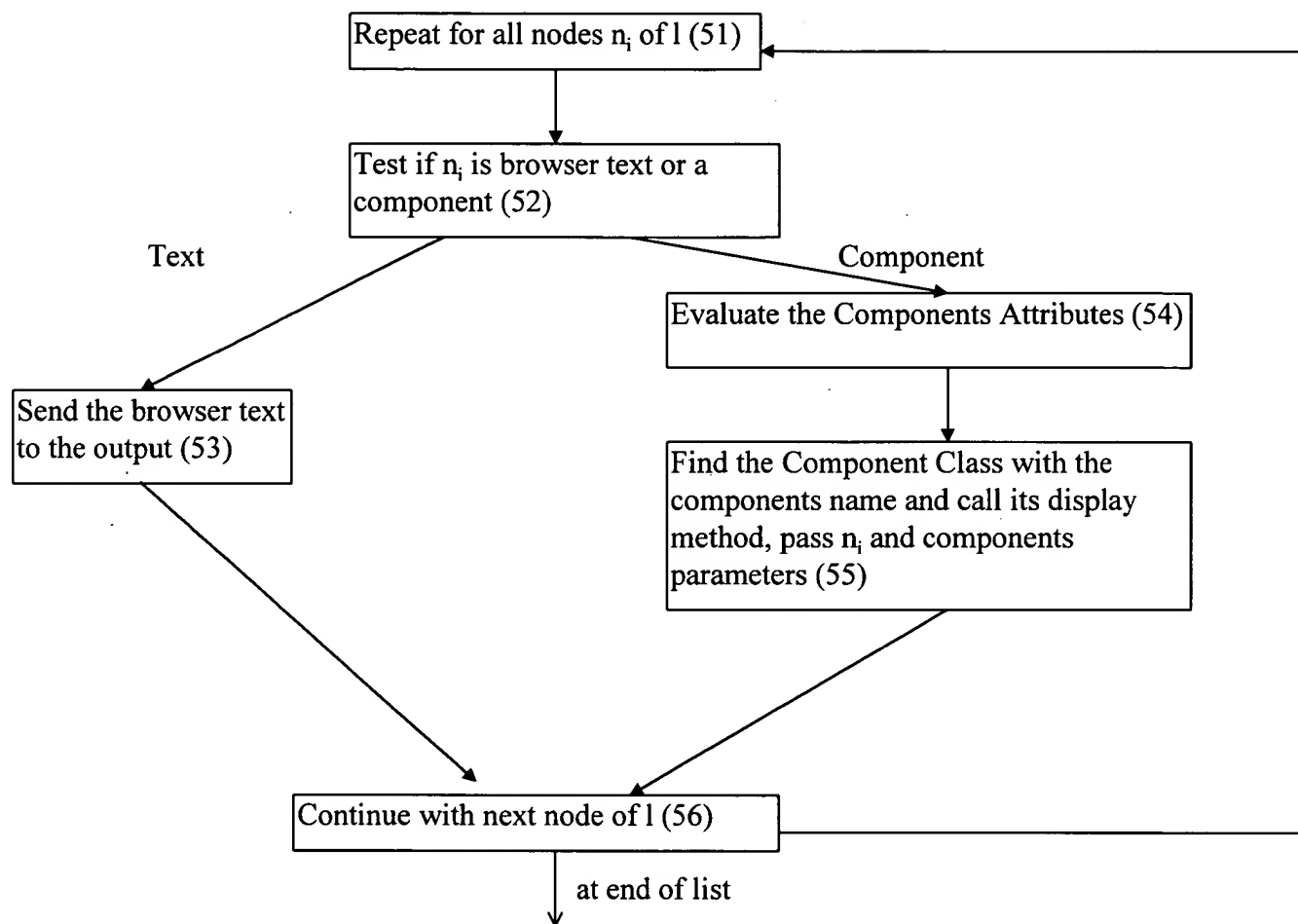


Fig. 9: Generation Algorithm

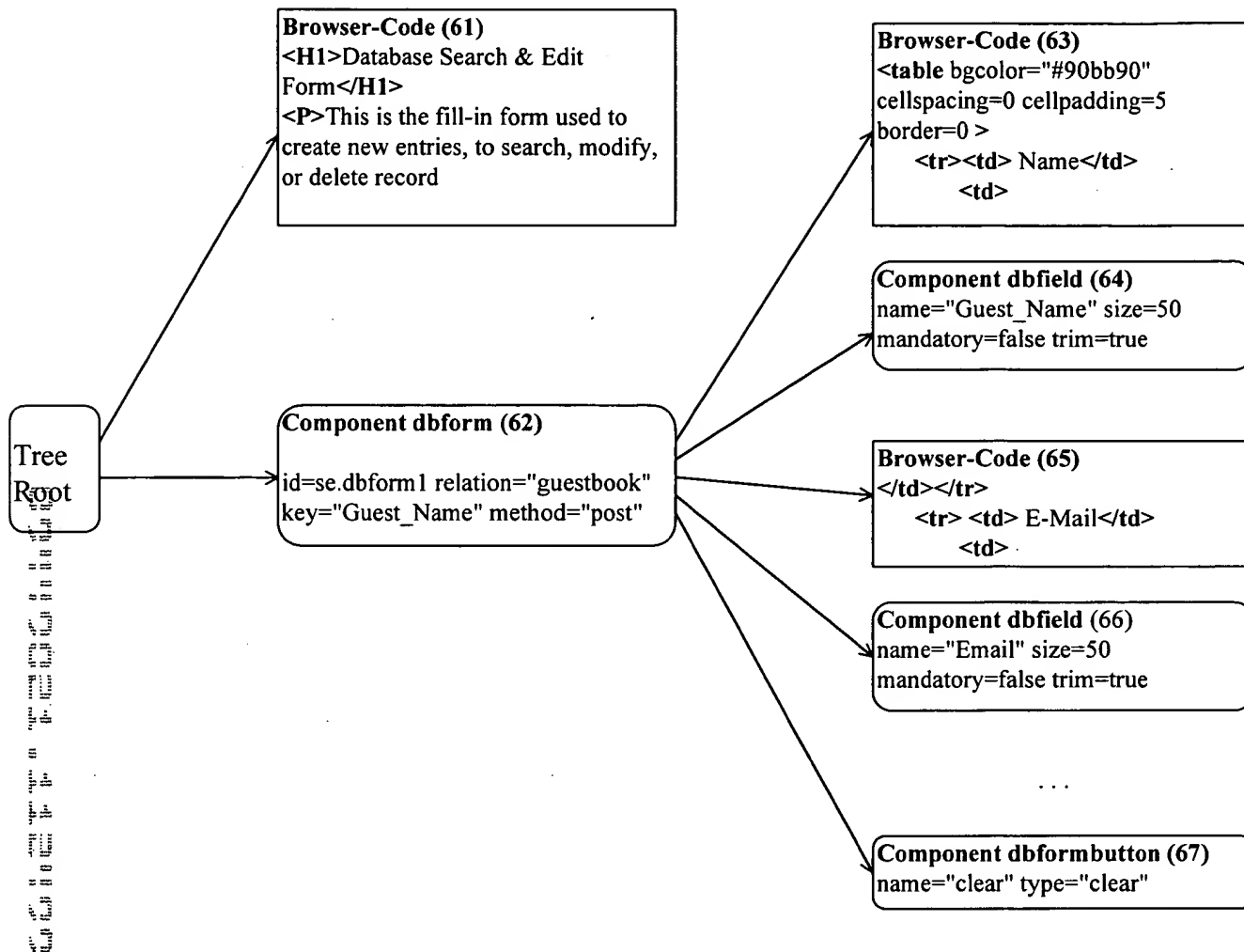


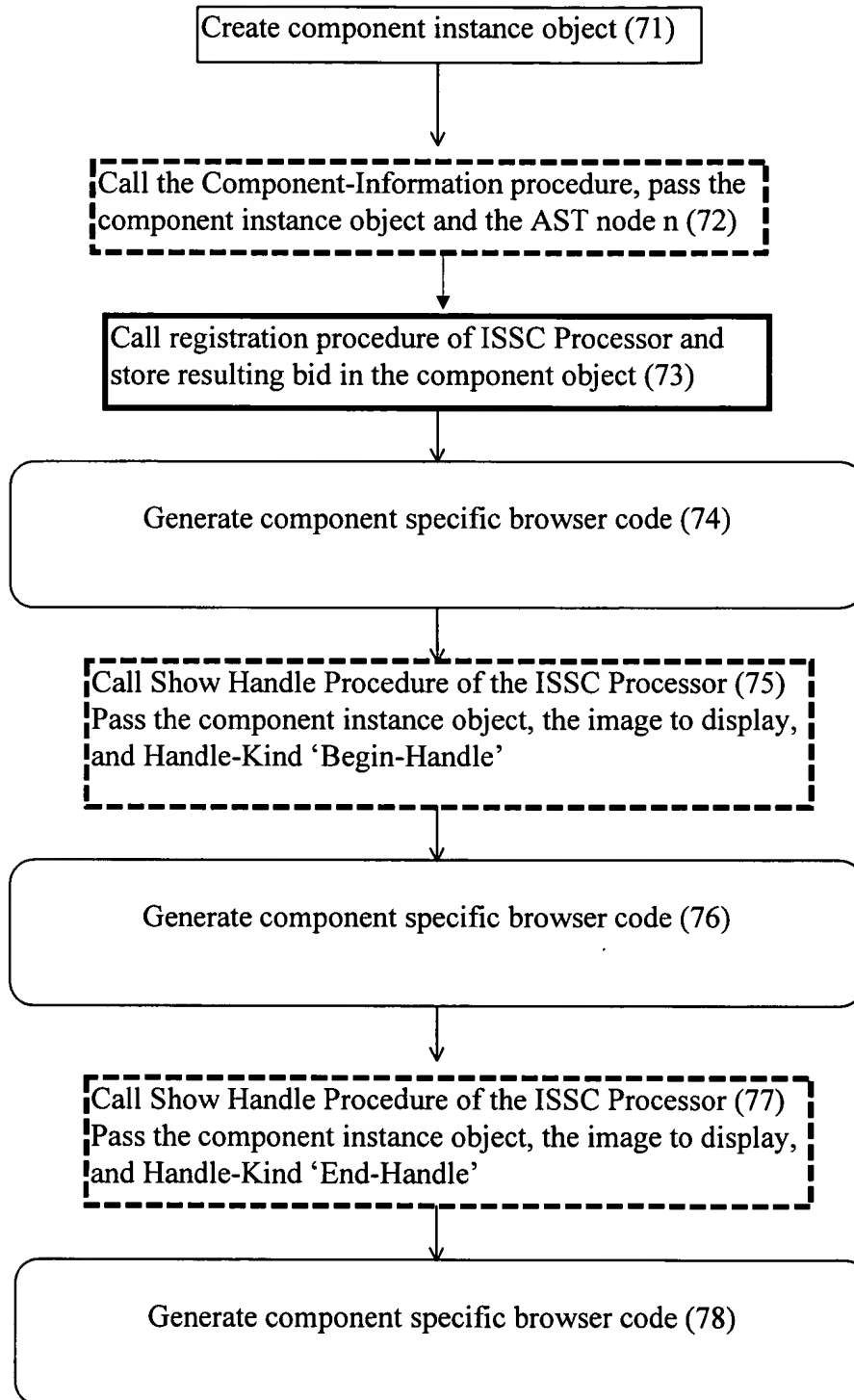
Fig. 10: Example Abstract Syntax Tree (AST)

## Display Method of a Component Class

Parameters:

AST Node: n

Parameter Values given in the Tag marking the Component



This sometimes is done implicitly by the programming language since display is a constructor

These calls can usually be inherited using an object oriented programming language.

Boxes with round corners define source code parts to be programmed per component

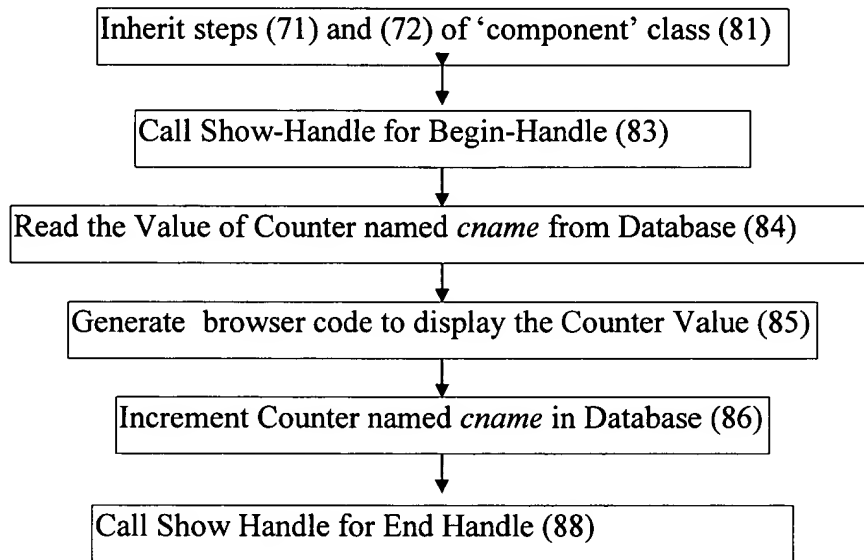
Boxes with broken lines call the component editor. These procedures work only if pages are displayed in edit-mode. To understand the component algorithm they

Fig.11: Display Method Algorithm Structure

## Sample Component Class: Counter

Inherit from Class 'Component'

Display Method, expects Counter Name as parameter *cname*



Steps (82) and (87) are not present because Drawing 9 shows an extended counter that has the same step numbers and uses (2) and (7)

Fig. 12: Example Counter Component Class

## Registration Procedure

Parameter:

Component Instance

Create bid as the kind of the component instance as string followed by the value of the global bid-counter and assign the bid to the bid field of the component instance. (91)

Increment the global bid-counter (92)

Insert the component instance into the list of listening components. (93)

Fig. 13: Registration Procedure

## Component Processing Algorithm

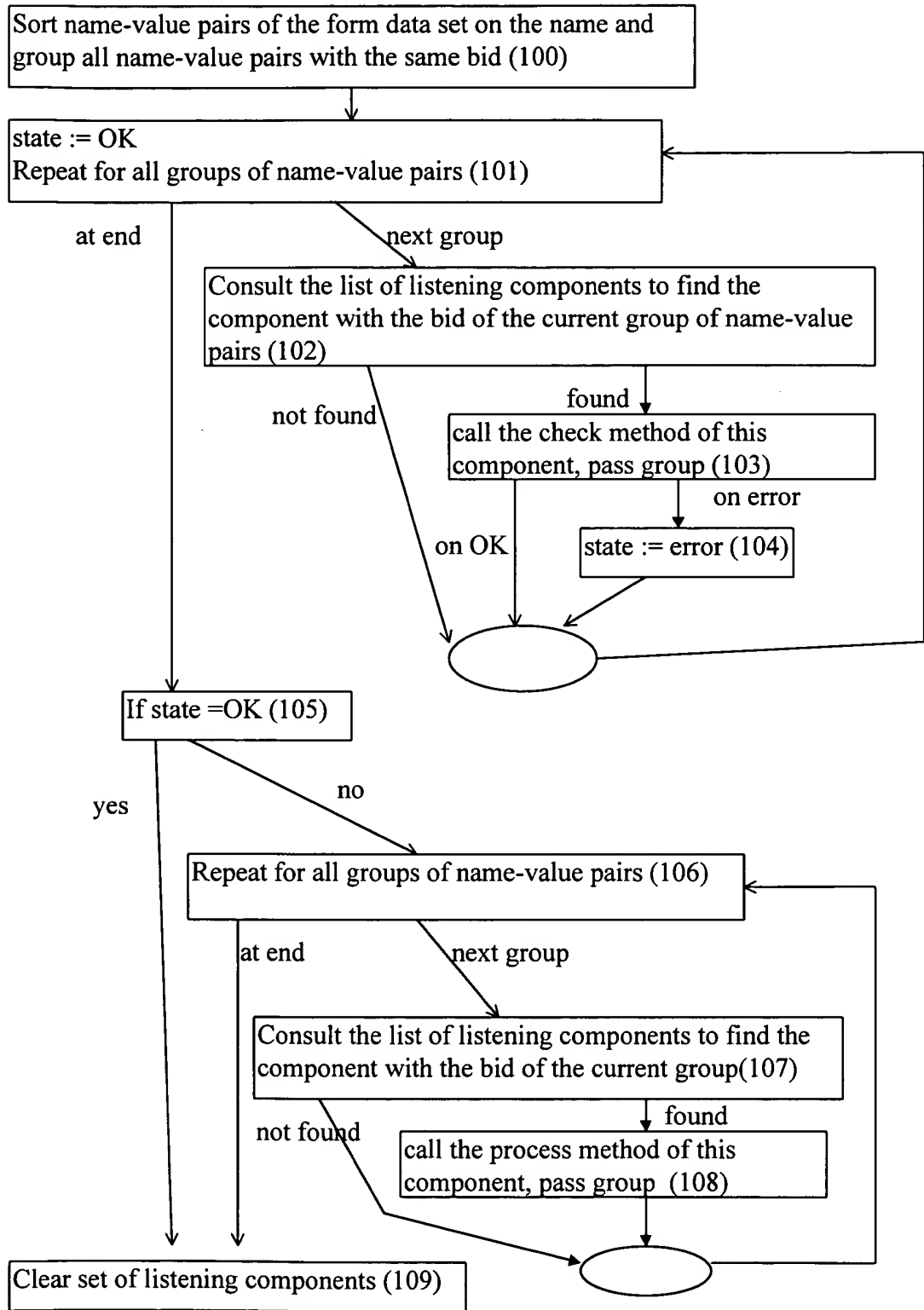


Fig. 14: Component Processing Algorithm

## Sample Component Class: Counter with Reset

Inherit from Class 'Component'

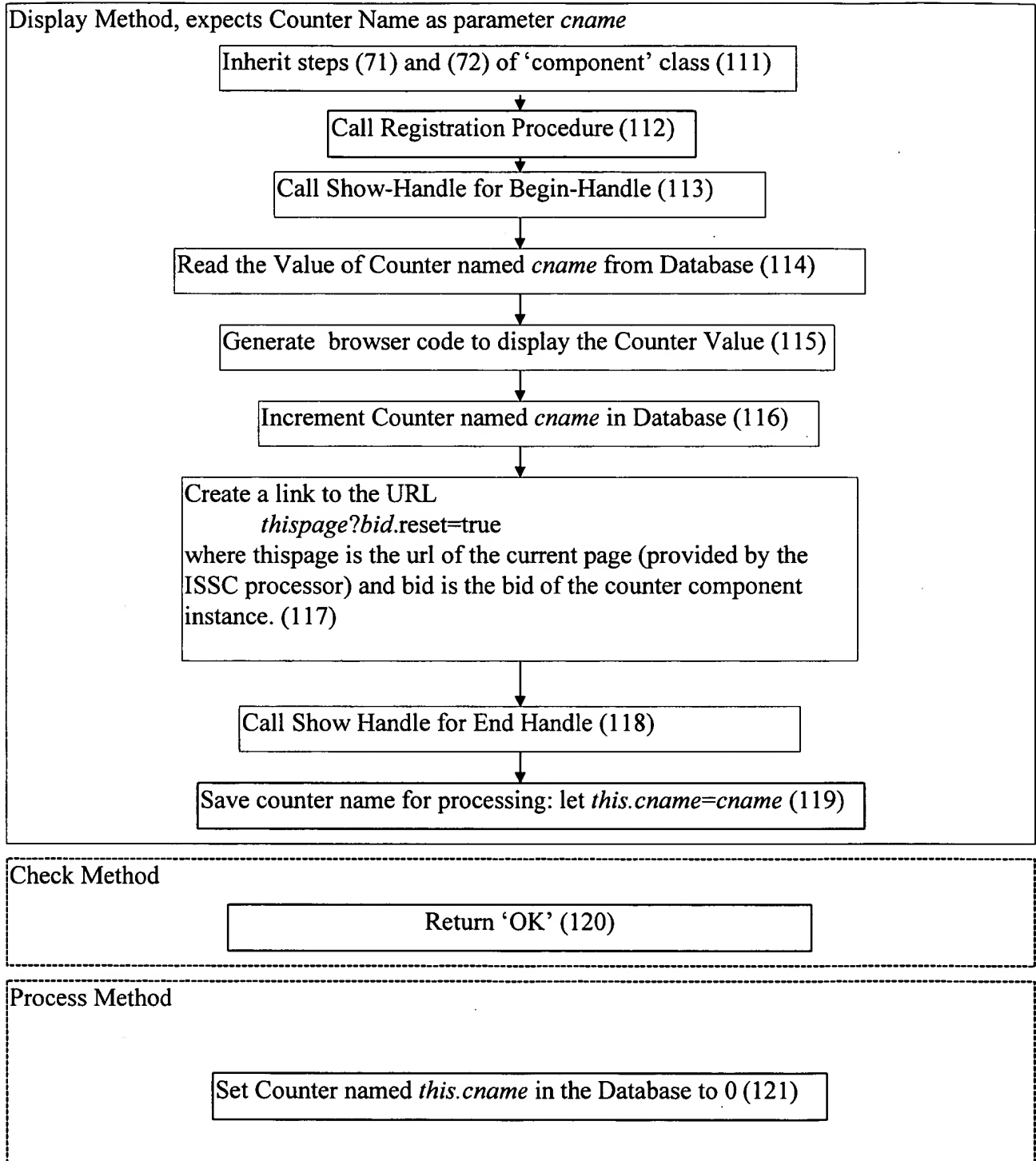


Fig. 15: Example Component Class for Counter with Reset

## Sample Component Class: dbinsertpanel

Inherit from Class 'Component'

Display Method, expects dbrelation as parameter

Inherit steps (71) and (72) of 'component' class (131)



Call Registration Procedure (132)



Call Show-Handle for Begin-Handle (133)



Set global variable curpanel to this, save the old value of curpanel  
let this.fieldlist = empty list  
let this.dbrelation = dbrelation (134)



Call Generate recursively for the content of dbinsertpanel (135)  
restore old value of curpanel



Call Show Handle for End Handle (136)

Check Method receives form data set group as parameter

Return 'OK' (140)

Process Method receives form data set group as parameter

Execute an SQL Insert Statement into the database relation this.relname. The field names are found in this.fieldlist and the field values in form data set group

Fig. 16: Example Component Class for dbinsertpanel



## Example Component Class: dbinsertfield

Inherit from Class 'Component'

Display Method, expects fieldname and fieldsize as parameter

Inherit steps (71) and (72) of 'component' class (151)

Call Show-Handle for Begin-Handle (152)

The global variable curpanel is the component object of the enclosing panel (153)  
Insert the fieldname into curpanel.fieldlist

Generate code for an HTML text input field. Field name is  
*curpanel.bid* ". *fieldname*  
and the fieldsize is given as parameter. (154)

Call Show Handle for End Handle (155)

Fig. 17: Example Component Class for dninsertfield

## Editor Structure

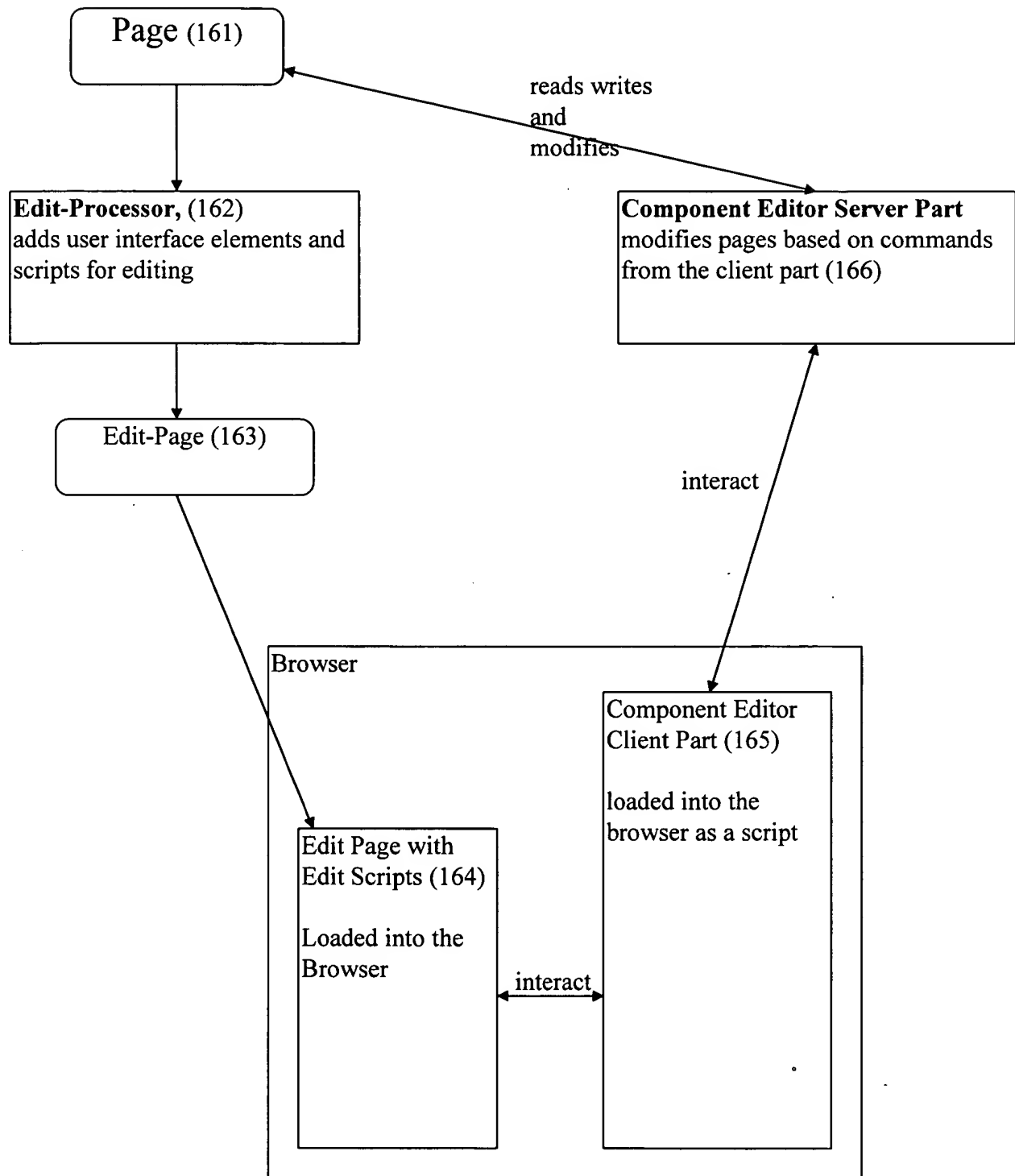


Fig.18: Editor Structure

## Component Editor Initialization Procedure

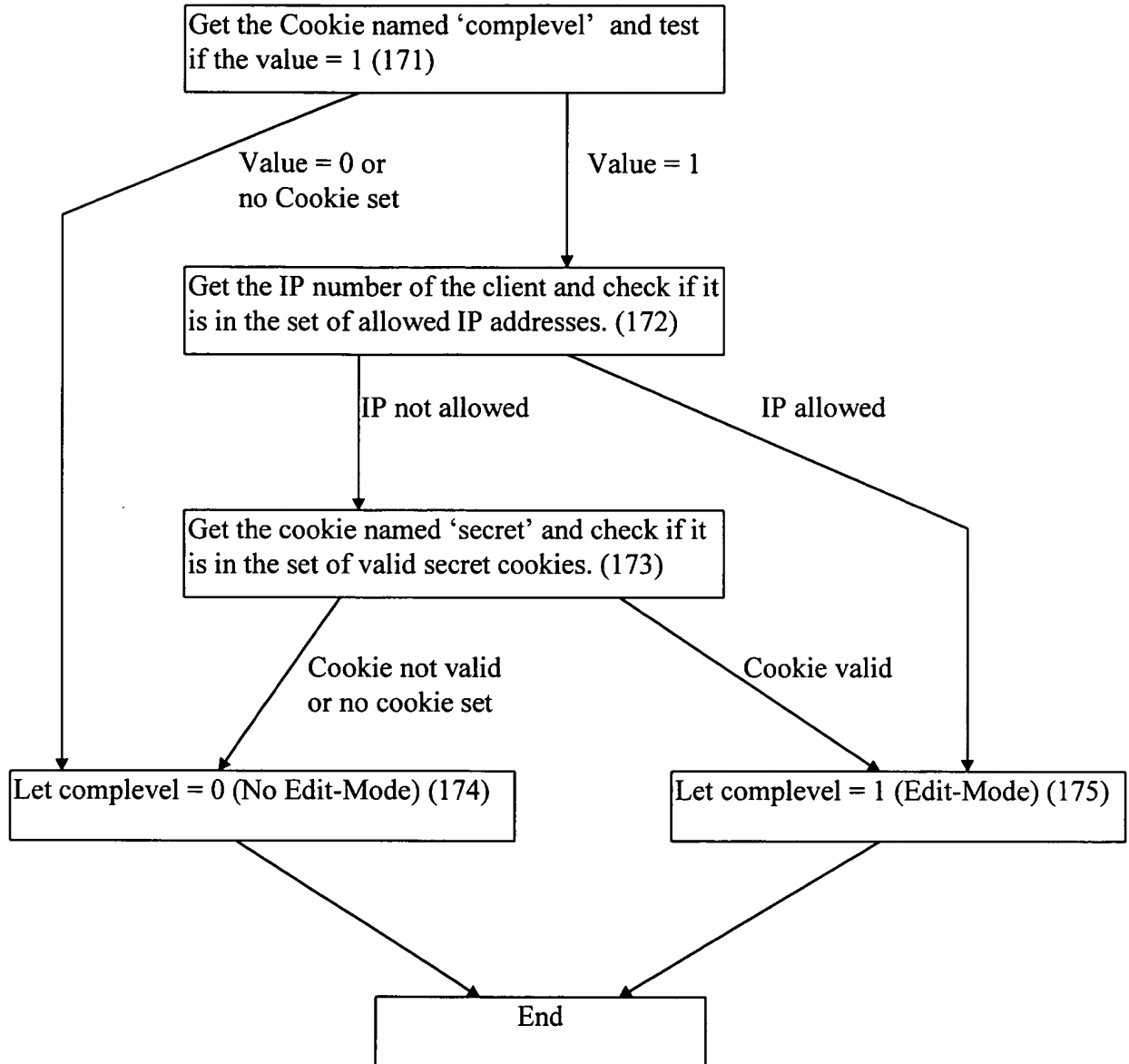


Fig.19: Component Editor Initialization Procedure

## Component Information Procedure

Parameters

Component node n of AST,  
Current component Instance

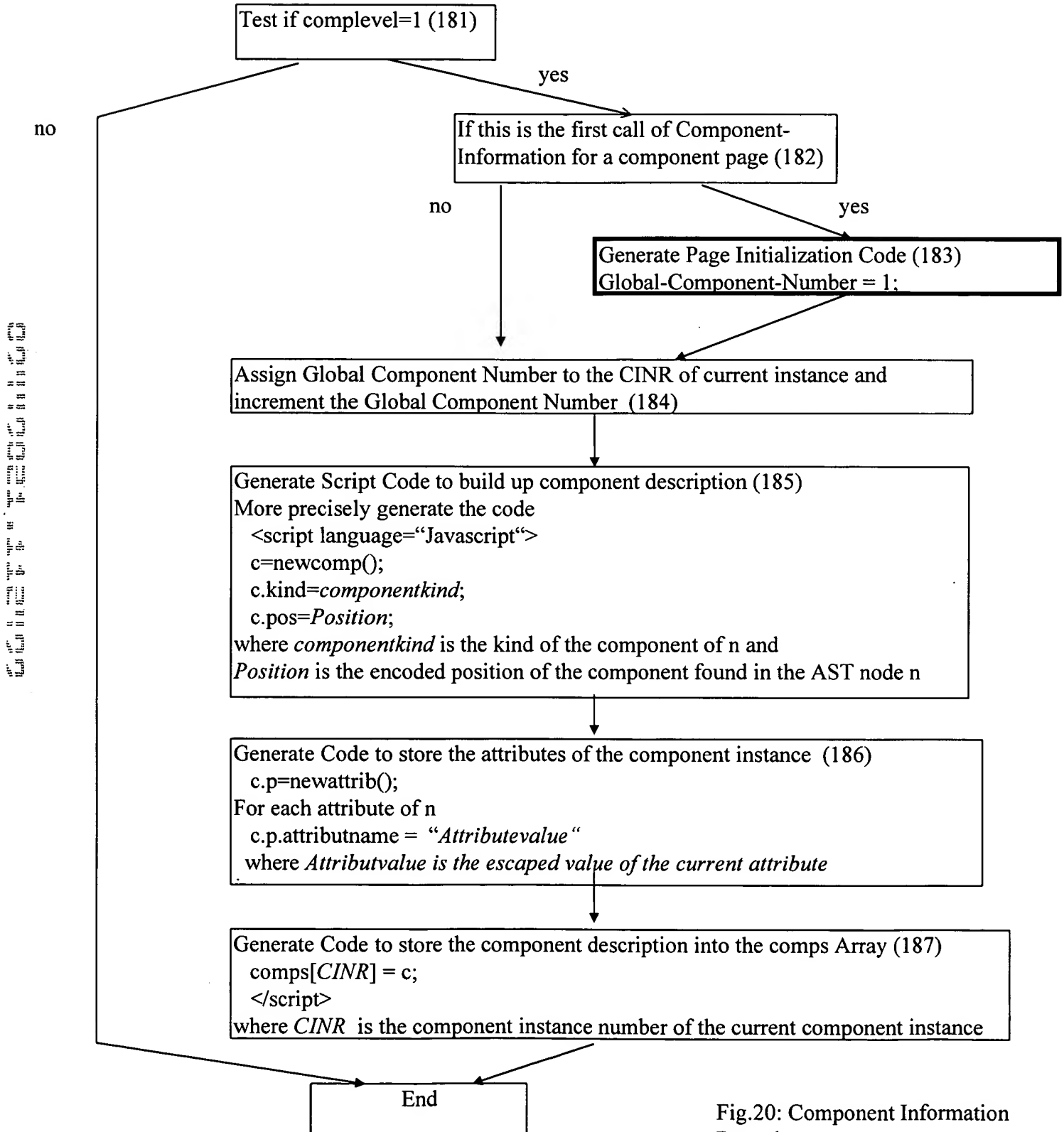


Fig.20: Component Information Procedure

## Show-Handle Procedure

### Parameters

Current Component Instance

Handle Kind: Begin-Handle or End-Handle

Handle Image: im

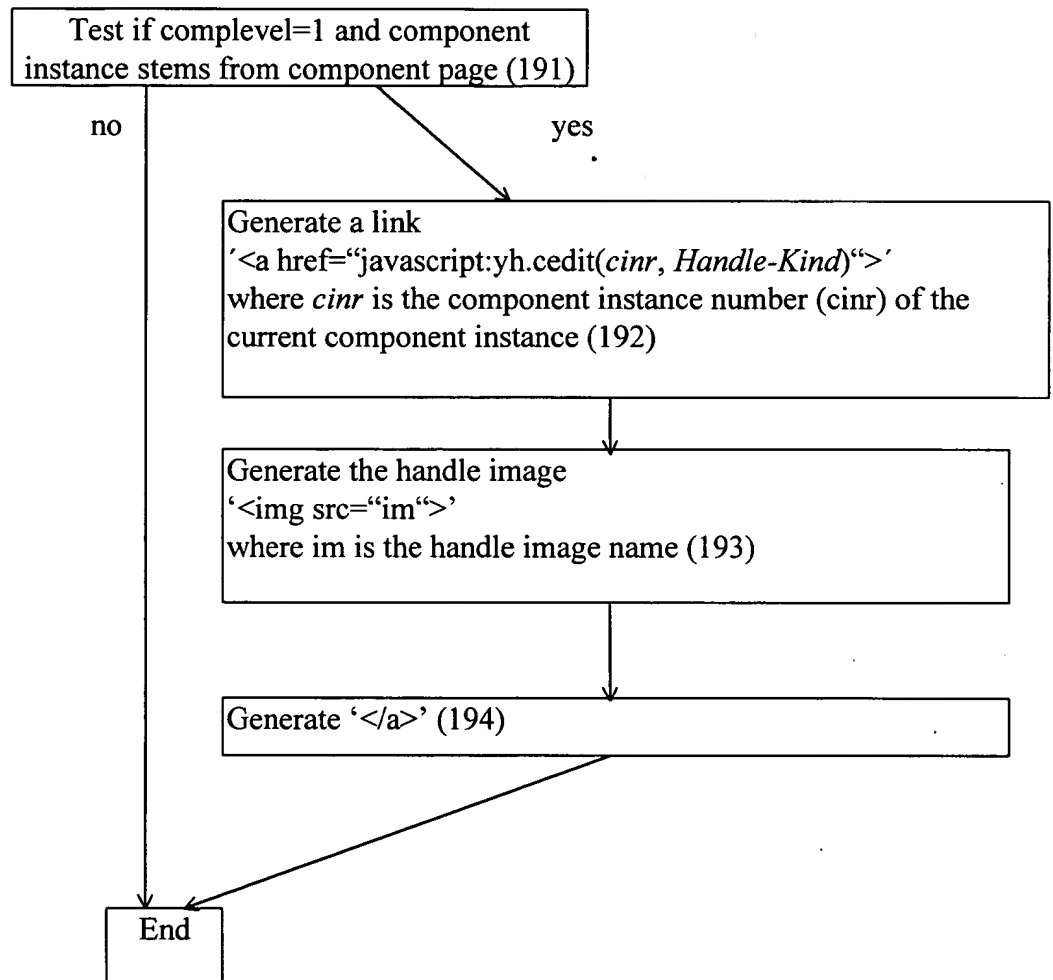


Fig.21: Component Editor Show Handle Procedure

## Generate-Page-Initialization-Code

Generate code to set the variable `yh` to point to the component editor control window. For example in javascript this is achieved by using the `open` function with the window name of the component editor control window.

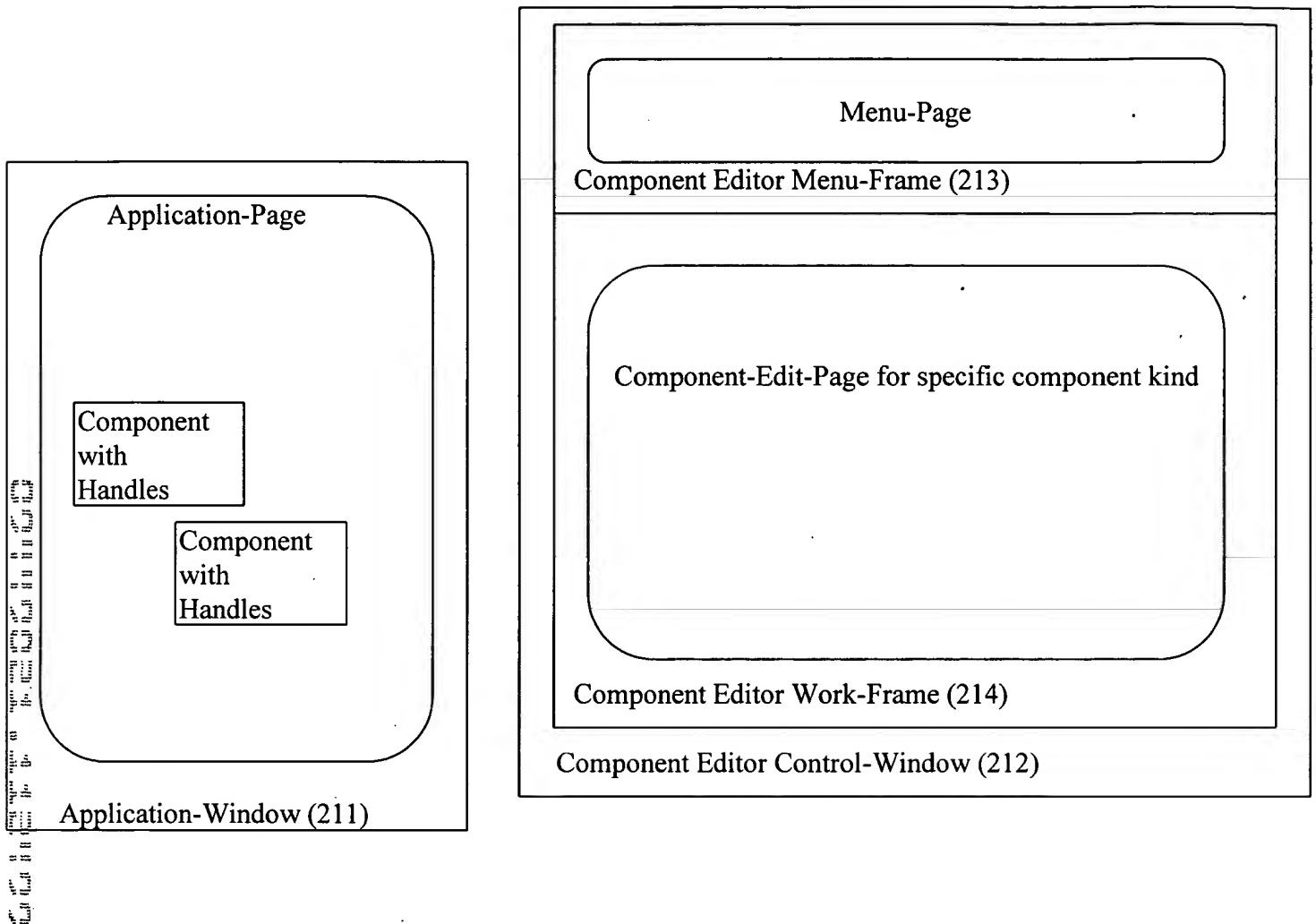
Generate code to store the own window handle in `yh.appwindow` so that the control window can access the Application-window. (201)



Generate definitions of helper functions necessary to create component descriptions. These are basically empty constructors to create the component description and attribute description objects. (202)

Fig.22: Generate Page Initialization Code

## Structure Component Editor Client Part



Fig,23: Structure Component Editor Client Part

## Component Editor Client Part Pages

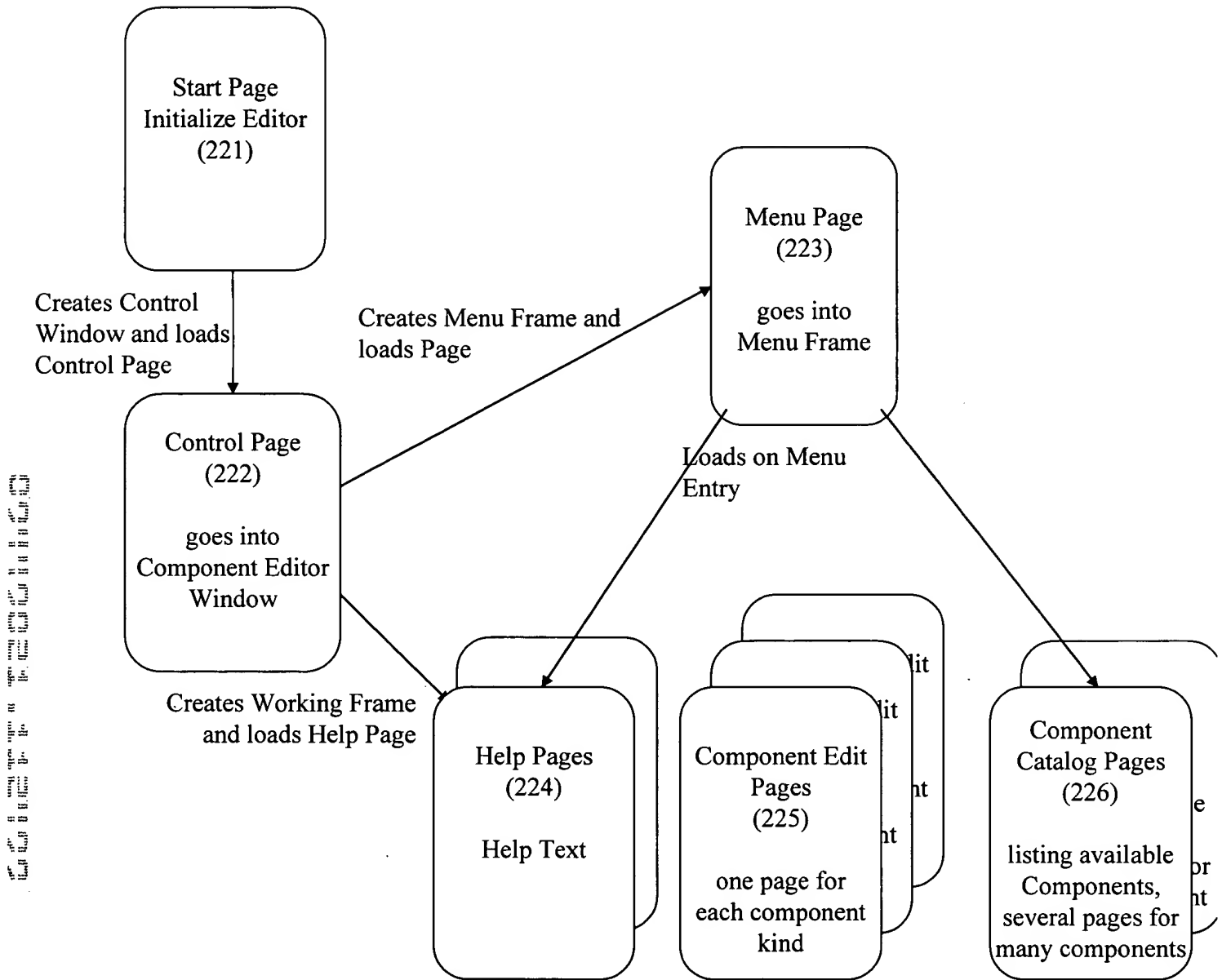


Fig.24: Component Editor Client Part Page Structure



## Component Editor Start Page

Call javascript open function to

- open up the component editor control window
- to load the component editor control page into it (231)

Message 'Component Editor Coming Up Please Wait' (232)

Links to Application Pages (233)

Fig. 25: Component Editor Start Page

## Component Editor Control Page

Insert Procedure Definition (241)

Cedit Procedure Definition (241)

Load Procedure (242)  
Set Complevel Cookie to 1

Unload Procedure (243)  
Set Complevel Cookie to 0

Frame set for Component Editor Window (244)

- Menu Frame with Menu Page
- Working Frame with initial Help Page

Fig. 26: Component Editor Control Page

## Insert Procedure

Parameter: Component Kind

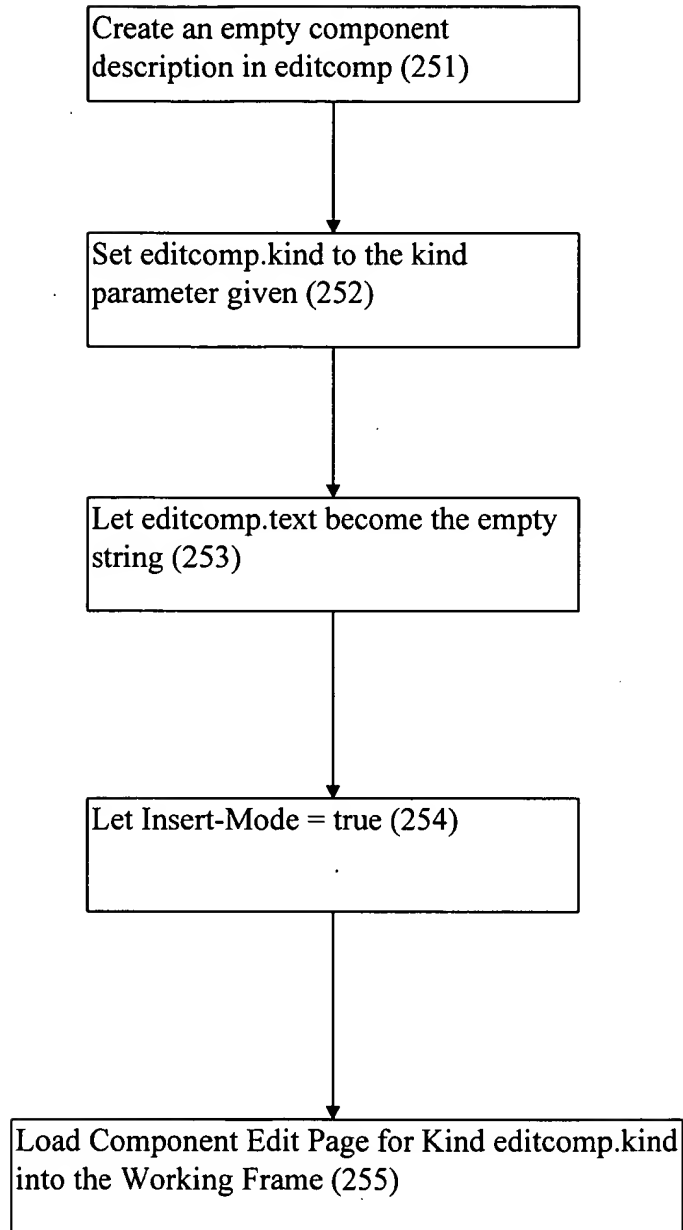


Fig. 27: Insert Procedure

## cedit Procedure

Parameters

Window: w

Component Instance Number: cn

Handle Kind (Begin or End Handle): hk

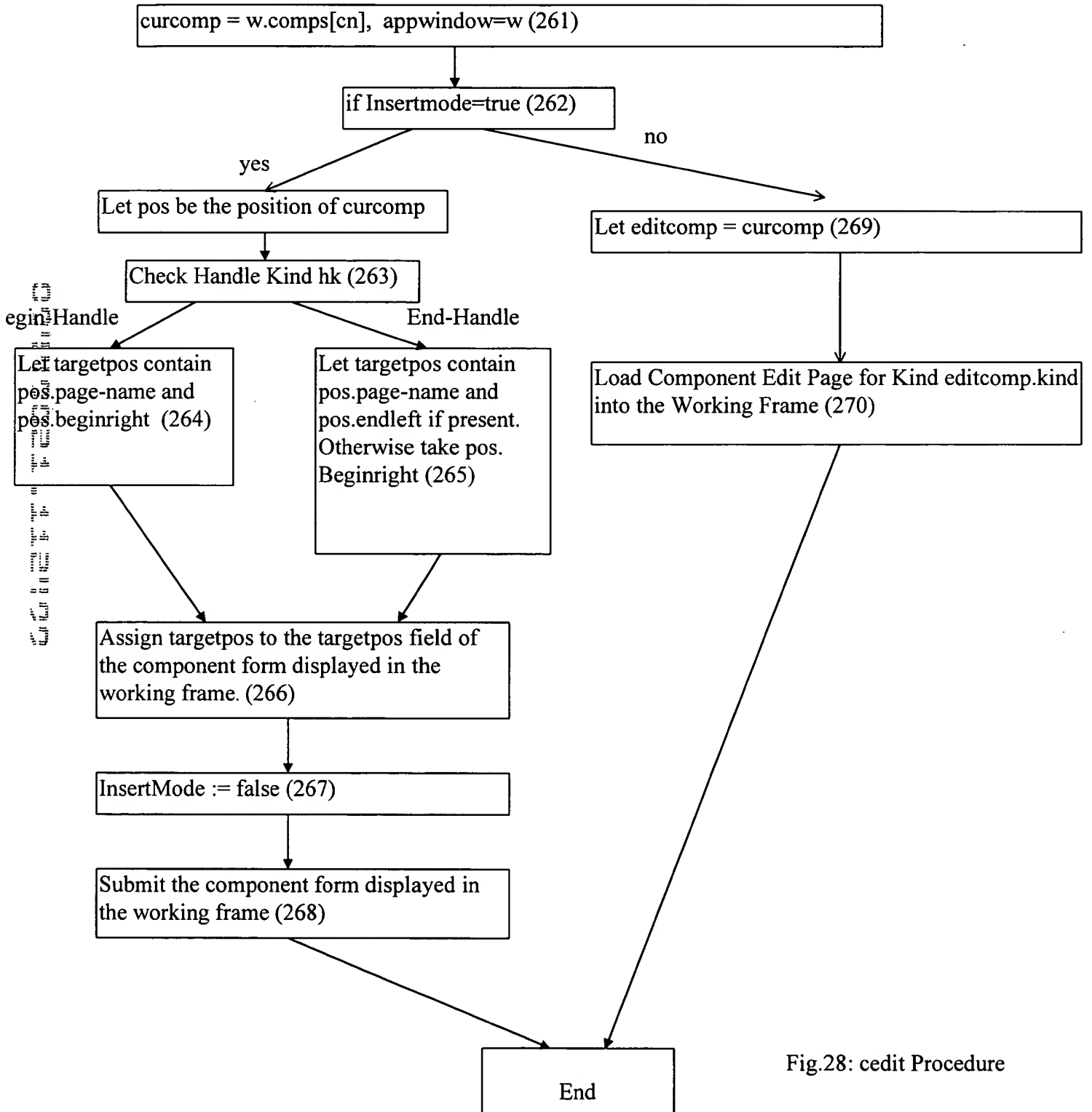


Fig.28: cedit Procedure

# Component Edit Page

HTML Form called Component Edit Form (281)

Form Fields one for each possible component attribute initially empty (282)

Hidden Component-Kind Field containing Component Kind (283)

Hidden Component Position Field initially empty (284)

Hidden Target Position Field initially empty (285)

Textarea for Component Content (286)

Submit Button(287)

Content Button(288)

Delete Button (289)

## Onload Procedure

Copy attribute values from top.editcomp.p into attribute form fields (290)  
Copy position of editcomp.p into position field

Fig.29: Component Edit Page

## Component Editor Server Part

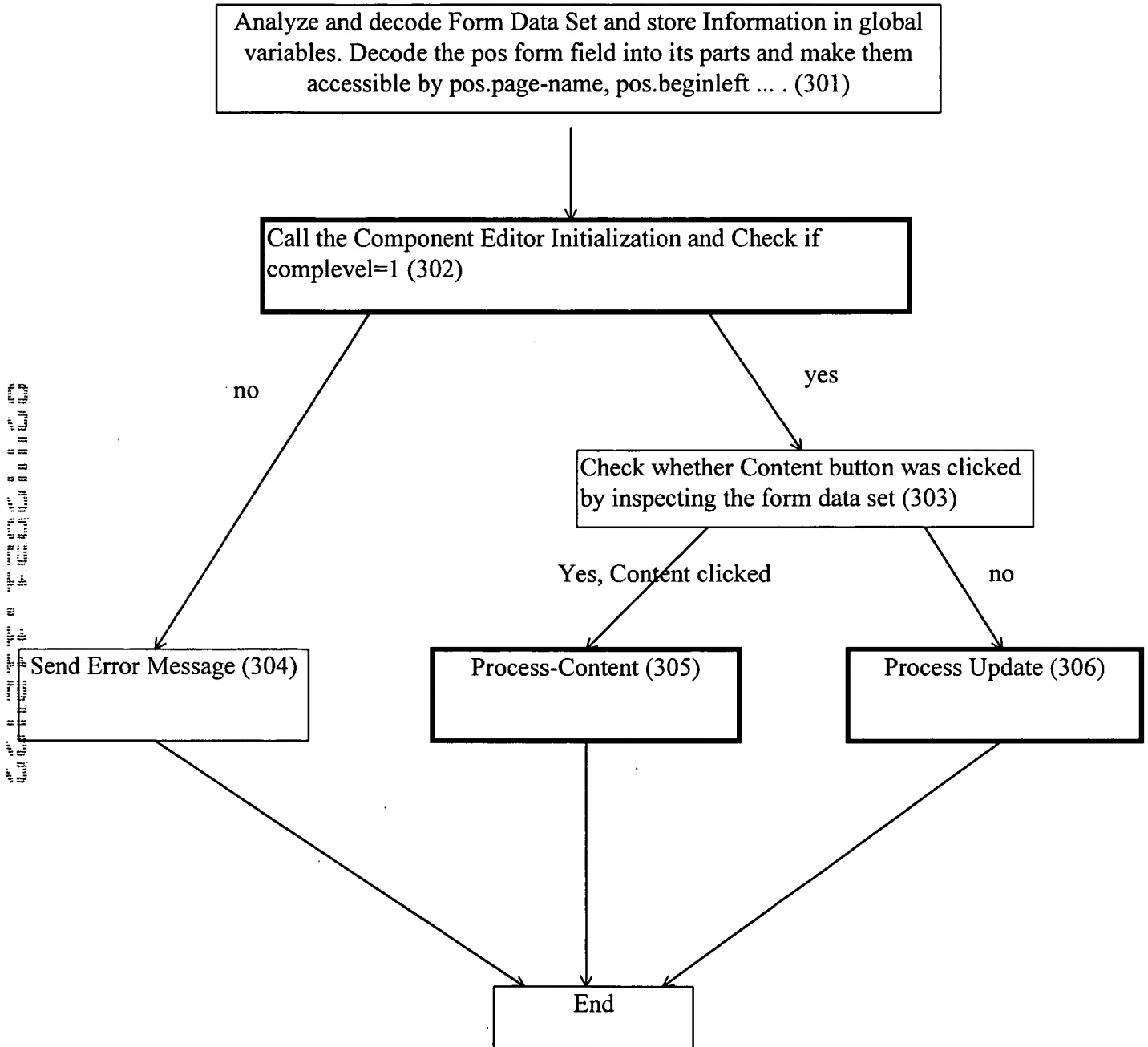


Fig.30: Component Editor Server Part

## Process-Content Procedure

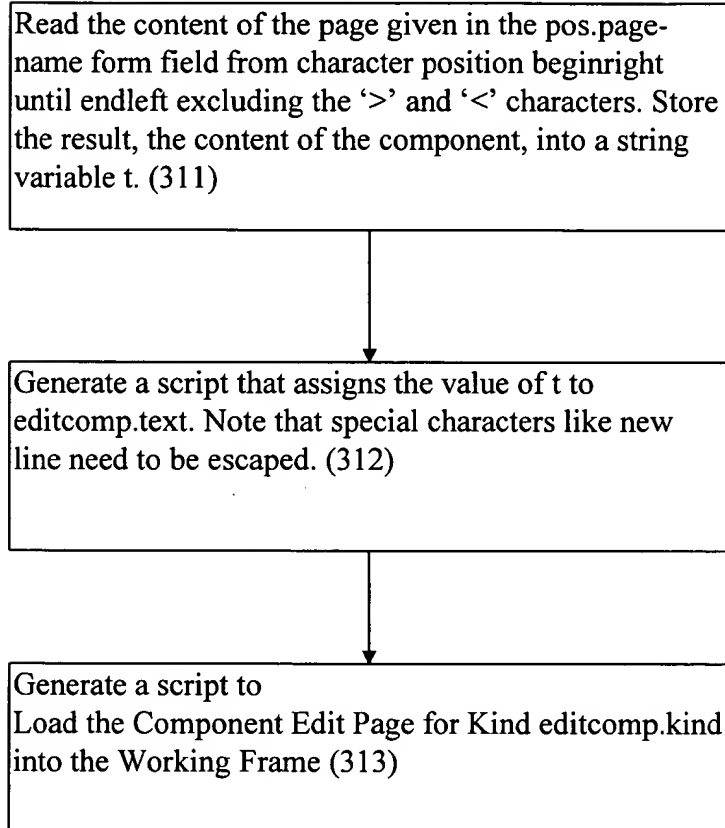


Fig.31: Process Content Procedure

## Process Update

Parameter:

Decoded Form Data Set of component form

Decoded position pos

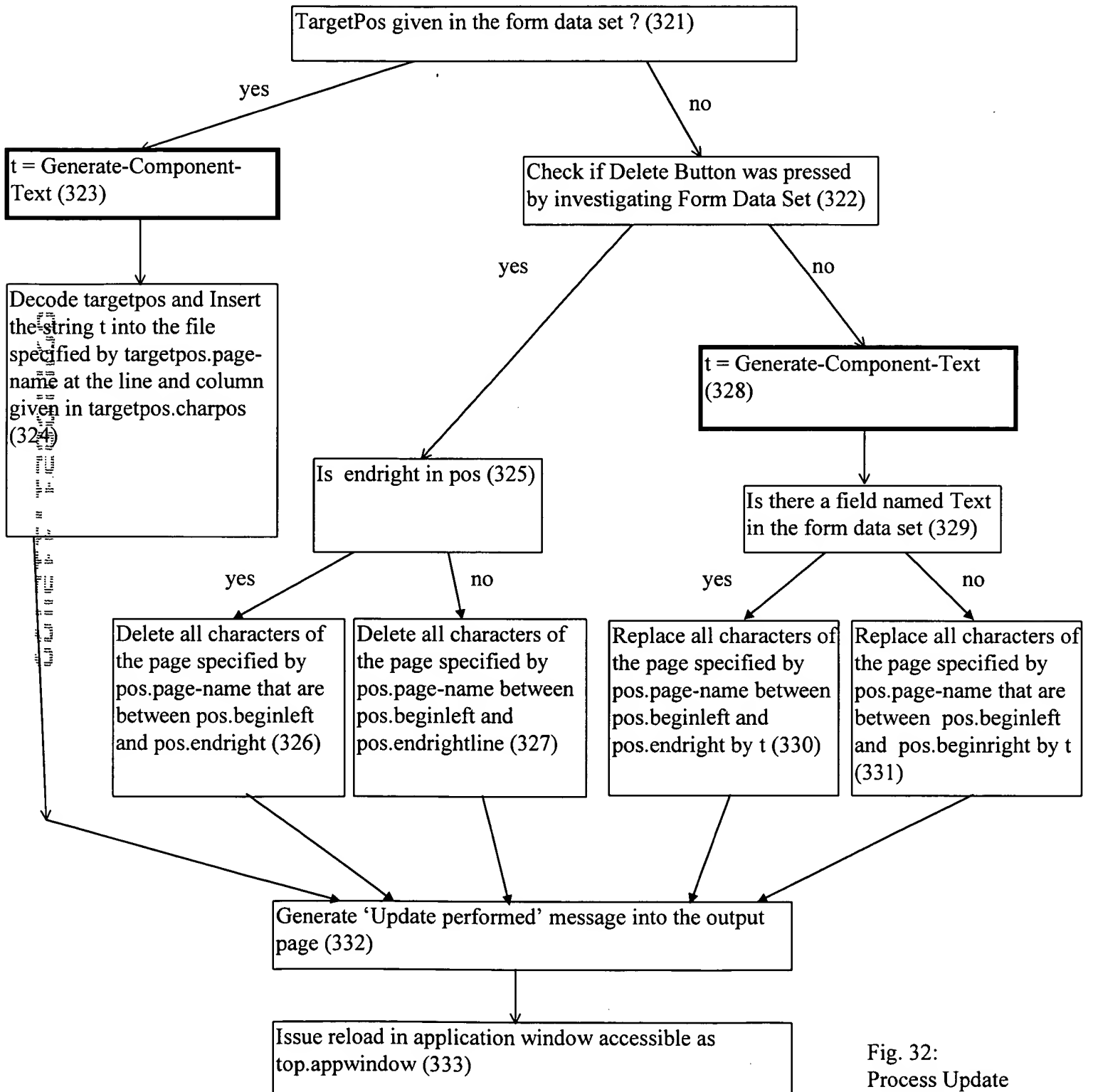


Fig. 32:  
Process Update  
Procedure



## Generate-Component-Text Procedure

Parameters:

Decoded Form Data Set of Component Form

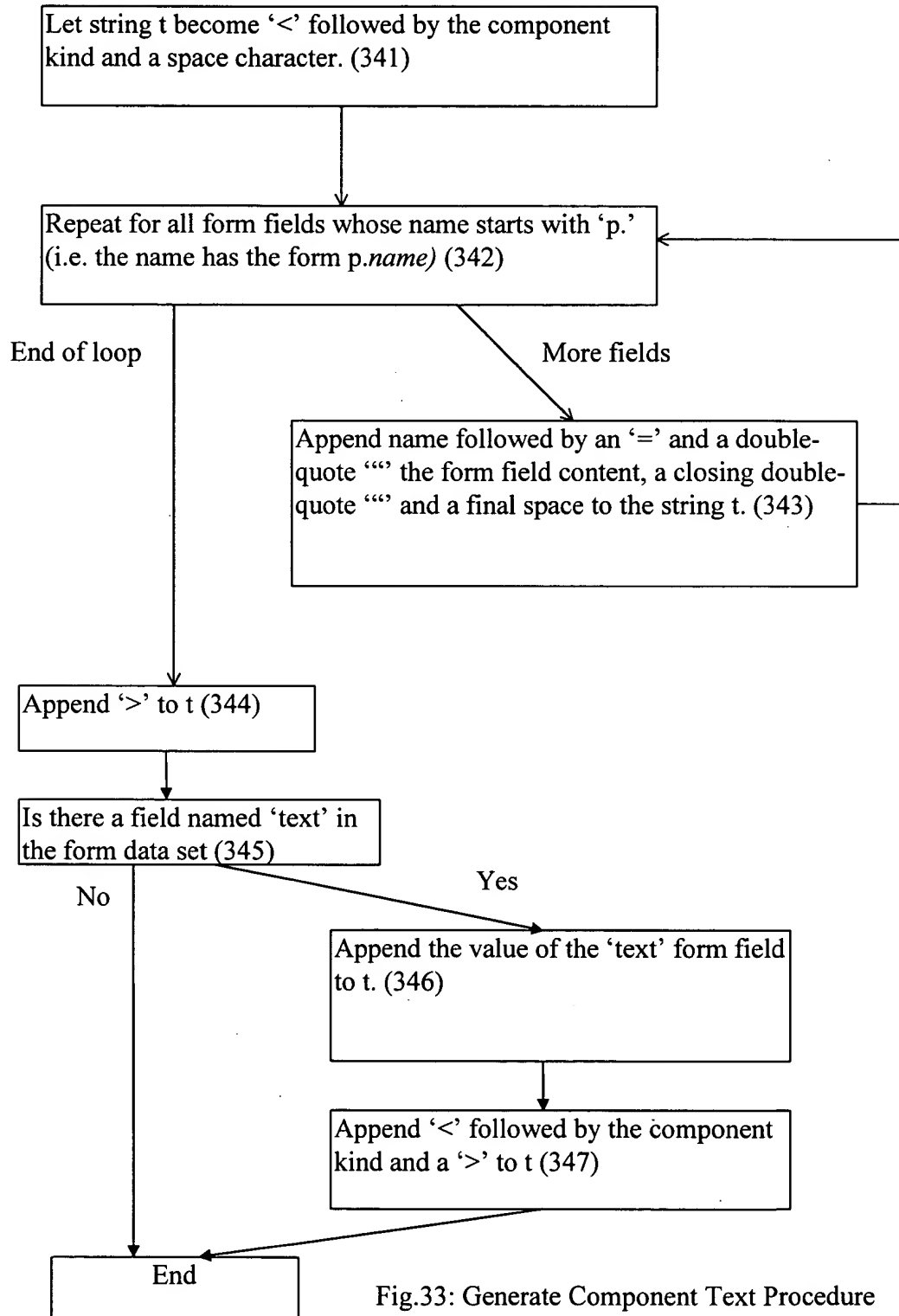


Fig.33: Generate Component Text Procedure

## Multi-Window Applications

To make the generation algorithm work for multiple windows replace step (93) of the registration subprocedure of the generation algorithm by

Insert the component instance into that list of listening components that belongs to the destination window. (351)

Insert Step (352) before step (101) of Fig. 14

Step (100) of Fig. 14

Let the list of listening components become the concatenation of all the lists of listening components of all windows. (352)

Step (101) of Fig. 14

Replace Step (109) of Fig. 14 by (step 353)

Clear the list of listening components that belongs to the destination window (353)

Fig.34: Mutli-Window Applications

## Persistent Components

(71) of Fig. 11 can be replaced by

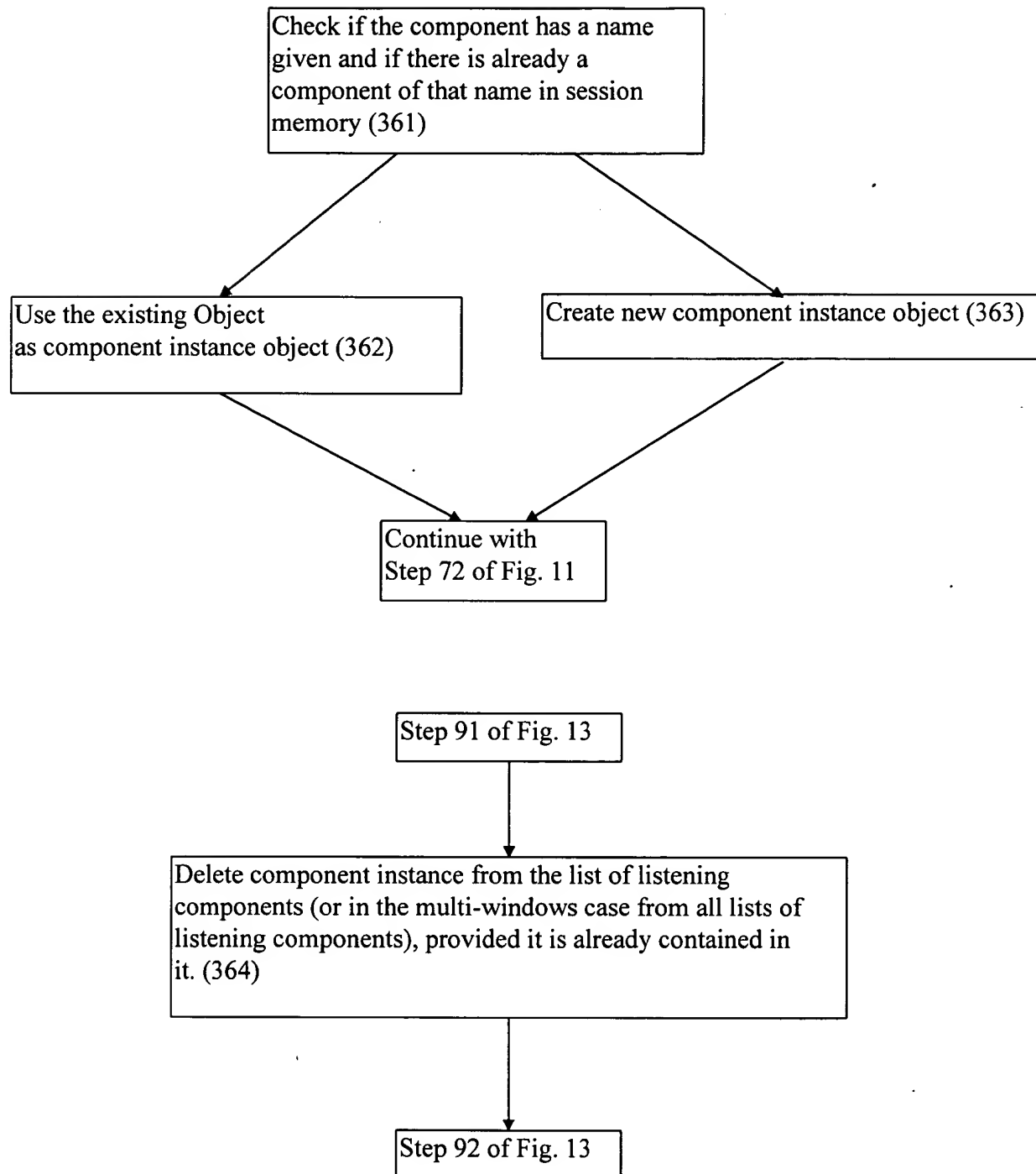


Fig.35: Persistent Components

## Session Less ISSC

Session variables like bid counter, and list of listening components become global server variables shared for all users.

Step 109 of page 114 becomes:

Remove all component objects from the list of listening components that were added longer than a fixed time-out value ago. (371)

Fig. 36: Session Less ISSC